

Plugin pro aplikaci Qtstalker

Qtstalker Plugin

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Jiří Dobeš**
Studijní program: B2647 Informační a komunikační technologie
Studijní obor: 2612R025 Informatika a výpočetní technika
Téma: **Plugin pro aplikaci Qtstalker
QTStalker Plugin**

Zásady pro vypracování:

Qtstalker je burzovní software vyvíjený pod licencí GNU GPL. Qtstalker disponuje běžnými nástroji pro technickou analýzu, indikátory, umožňuje import EOD dat z CSV souborů, případně obsahuje plugin pro download EOD dat ze zdrojů jako je Yahoo. Program umožňuje pracovat s akcemi, komoditami i spready. Realtime obchodování program v současné době nepodporuje. Součástí programu je i prostředí pro backtestování vlastních obchodních systémů.

Cílem této bakalářské práce je implementovat plugin, umožňující stahování dat z pražské burzy cenných papírů (SPAD, KOBOS, RMS). Vhodné by bylo rozšířit i možnosti stávající aplikace pro intradenní (realtimové) obchodování.

Seznam doporučené odborné literatury:

Paul Deitel, C++ How to Program, Prentice Hall; 7 edition, 2009, ISBN 978-0136117261
Bjarne Stroustrup, Programming: Principles and Practice Using C++, Addison-Wesley Professional; 1 edition, 2008, ISBN 978-0321543721
Anatoly B. Schmidt, Financial Markets and Trading: An Introduction to Market Microstructure and Trading Strategies, Wiley; 1 edition, 2011, ISBN 978-0470924129

<http://www.bcpcp.cz>

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Michal Krumník**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



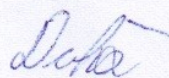
doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 4. května 2012

A handwritten signature in blue ink, appearing to be 'D. K.', is written above a dotted line.

.....

Chtěl bych poděkovat všem, kteří mi jakoukoliv mírou pomohli při tvorbě této bakalářské práce, především pak vedoucímu této práce Ing. Michalovi Krumníkovi za ochotnou pomoc při řešení problémů.

Abstrakt

Tato bakalářská práce pojednává o tvorbě pluginů pro aplikaci Qtstalker. Obsahuje obecný postup, jak vytvořit plugin pro tuto aplikaci. Dále pak obsahuje popis dvou konkrétních pluginů vytvořených v rámci této bakalářské práce. Jedná se o plugin umožňující stahování dat z pražské burzy a o plugin pro realtime sledování obchodování pomocí služeb Yahoo! Finance. Dále práce obsahuje informace o pražské burze cenných papírů.

Klíčová slova: Qtstalker, QT, PSE, Yahoo, pražská burza, finance, cenné papíry

Abstract

This bachelor thesis deals with the creation of plugins for the application Qtstalker. It contains general instructions how to create a plugin for this application. Then describes two specific plugins developed within this thesis. This is a plugin that allows downloading of data from the Prague Stock Exchange and the plugin for realtime monitoring of trading using Yahoo! Finance services. This thesis then includes information about the Prague Stock Exchange.

Keywords: Qtstalker, QT, PSE, Yahoo, Prague Stock Exchange, finance, securities

Seznam použitých symbolů a zkratek

AOS	- Automatizovaný obchodní systém.
API	- Application Programming Interface. Rozhraní pro programování aplikací.
BSD	- Berkeley Software Distribution. Licence pro svobodný software.
CME	- Chicago Mercantile exchange
CSV	- Comma-separated values. Souborový formát pro výměnu tabulkových dat, kde jednotlivé hodnoty jsou odděleny pomocí čárky.
CVS	- Concurrent Version Systém. Systém sloužící ke správě verzí projektu.
EOD	- End of day. Označení pro data, která jsou dostupná na konci obchodního dne.
FTP	- File transfer protocol. Protokol pro přenos souborů.
GNU	- Svobodný operační systém typu Unix, který však neobsahuje žádný originální kód Unixu. V současné době však používá jiná svobodná jádra.
GNU GPL	- GNU General Public License. Všeobecná veřejná licence pro svobodný software.
GUI	- Graphic user interface. Grafické uživatelské rozhraní.
ISIN	- International Securities Identification Number
NYSE	- New York Stock Exchange. Americká burza se sídlem v New York City
OS	- Operační systém.
PSE	- Prague stock Exchange. Burza cenných papírů Praha.
SVN	- Apache Subversion. Systém pro správu verzí systému.
URL	- Uniform Resource Locator. Jednotný lokátor zdrojů.

Obsah

1. Úvod.....	2
2. Aplikace pro práci s investičními nástroji	3
2.1 Open Source	3
2.2 Freeware	4
2.3 Komerční software	5
3. Zdroje dat a datová struktura.....	7
3.1 Burza cenných papírů Praha	8
3.2 Yahoo! Finance	9
3.3 Google Finance	10
4. Burza cenných papírů Praha	11
4.1 Burzovní den	11
4.2 Automatizovaný obchodní systém (AOS).....	12
5. Seznámení s programem Qtstalker a ostatními prostředky.....	13
5.1 Qtstalker	13
5.2 Ta-Lib	14
5.3 Berkeley-DB	14
5.4 QT (knihovna)	15
5.5 QT Creator	15
6. Obecná tvorba pluginu pro aplikaci Qtstalker.....	17
6.1 Seznámení s třídou QuotePlugin	19
6.2 Postup při vytváření pluginu	21
7. PSE plugin.....	25
7.1 Funkčnost	25
7.2 Zpracování dat.....	25
7.3 Downloader	26
7.4 Problémy během vývoje.....	26
8. YahooRealtime plugin.....	27
8.1 Funkčnost	27
8.2 Zpracování dat.....	27
9. Závěr.....	28
10. Reference.....	29
Přílohy	
A Tabulka parametrů pro požadavek na službu Yahoo! Finance.....	31
B CD.....	33

1. Úvod

Burza je jednou ze základních součástí kapitálového trhu a díky ní je možné obchodovat s různými investičními nástroji, jakými jsou investiční cenné papíry, cenné papíry kolektivního investování, nástroje peněžního trhu a deriváty.

Tato bakalářská práce se zabývá problematikou sledování obchodování na burzách s investičními nástroji, především pak na unixových systémech. Pro tuto platformu existuje několik programů, určených pro získávání dat týkajících se investičních nástrojů. Následující kapitola je věnovaná burzovnímu softwaru, jeho vlastnostem a srovnání několika představitelů.

Další část je určena k seznámení s pražskou burzou, službou Yahoo Finance! a Google finance a fázemi obchodování. V této kapitole jsou taktéž obsaženy informace o poskytovaných datech a jejich typickém formátu.

Následující kapitola obsahuje jak samotné seznámení s programem Qtstalker, tak seznámení s důležitými prostředky pro vývoj. Podkapitola zabývající se prostředím Qt Creatoru si klade za cíl ukázat přednosti tohoto prostředí, stejně tak, jako seznámit s použitím například slotů a signálů. Pro danou problematiku této práce byl vybrán program Qtstalker, jehož funkčnost a možnosti rozšíření včetně informací o potřebných knihovnách a jejich verzích objasní další z podkapitol.

Jedním z hlavních cílů této práce je vytvořit plugin, který umožní uživatelům stahovat data z pražské burzy do programu Qtstalker a dále pak rozšířit program o možnost reálného sledování obchodování. Reálná funkčnost je dostupná pouze pro data získaná z Yahoo! Finance, jelikož pražská burza neposkytuje realtime data zdarma. Navíc by tato práce měla sloužit i jako návod pro budoucí další rozšíření programu Qtstalker o jiné pluginy.

Kapitola zabývající se konkrétními pluginy obsahuje popis jejich funkčnosti, popis zpracování dat a eventuální problémy při jejich vývoji. V závěru práce bude zhodnocena použitelnost aplikace a nastíněna její další rozšiřitelnost.

2. Aplikace pro práci s investičními nástroji

Aplikací, které se zabývají sběrem a prezentací dat z oblasti burzovního obchodování, je na trhu poměrně mnoho. Ne všechny jsou však natolik propracované, aby uspokojily i náročnější uživatele. Většina z nich zvládá získávání dat z různých webových služeb nebo souborů a jejich překreslení do grafů. Ty pokročilejší navíc zvládnou různé exporty dat, přidávání vlastních indikátorů, kreslení do grafů, upozornění při překročení nastavené hodnoty, prodej/nákup apod. Dalším podstatným rozdílem je způsob licencování těchto programů. Cena komerčních produktů se pohybuje v řádu desítek až stovek amerických dolarů. Programů zdarma není takové množství jako těch komerčních, na druhou stranu i zde lze nalézt několik kvalitních zástupců. Jednou z hlavních výhod open source aplikací je možnost jejich libovolného upravení dle vlastních požadavků. Programy s uzavřeným kódem lze v některých případech taktéž rozšiřovat, ale pouze do míry, jakou poskytuje autor programu. Porovnat několik zástupců aplikací pro práci s investičními nástroji lze v tabulce 1.

2.1 Open Source

Jde o bezplatný software s otevřeným zdrojovým kódem. To znamená, že uživatel si může prohlížet zdrojové kódy programu a libovolně je upravovat. Výhodou je, že chyby v kódu může hledat větší skupina lidí. Na druhou stranu se tento kód poskytuje i případným útočníkům.

Qtstalker

Jedná se o aplikaci pro unixové systémy. Pokud se uživatel rozhodne pro instalaci ze zdrojových kódů, pak je zprovoznění programu poněkud komplikované oproti ostatním zástupcům. Program nabízí běžné prostředky pro technickou analýzu, práci s indikátory, export/import dat. V základní instalaci obsahuje prostředky pro získání EOD dat z různých služeb a import dat z CSV souborů. Jako jediný z programů zmíněných v této kapitole nepodporuje realtime sledování. Je velmi dobře rozšiřitelný díky použitému systému pluginů. Je vyvíjen v jazyce C++.

JStock

Obdobně jako Qtstalker umožňuje práci s indikátory a běžnou analýzu. Poskytuje registrovaným uživatelům cloudový prostor pro ukládání dat, takže uživatelé mají přístup ke svým datům prakticky odkudkoliv. JStock je dostupný pro unixové systémy, MS Windows, Mac OS X a Solaris. Data získává ze služby Yahoo! Finance a podporuje trhy 23 zemí. Umožňuje také automatické odesílání SMS zpráv v případě překročení nastavené hranice ceny dané položky. Aplikace je psána v jazyce Java.

Grism

Funkčně zaostává za aplikacemi Qtstalker a JStock. Umožňuje pouze zobrazovat stažené hodnoty a jejich vykreslení do grafu. Graf lze exportovat jako obrázek ve formátu PNG. Oproti Qtstalkeru ale disponuje funkcí pro realtime sledování obchodování. Data získává stejně jako JStock ze služby Yahoo! Finance. Podporované OS jsou MS Windows, Linux a Mac OS X. Jazykem pro vývoj tohoto softwaru je Ruby.

2.2 Freeware

Stejně jako u open source se jedná o bezplatný software, ovšem s tím rozdílem, že zdrojové kódy nejsou běžně přístupné, a i samotné jejich získávání ze spustitelných souborů je nelegální. Přesto, že se jedná o bezplatný software, občas autoři dávají možnost uživatelům v případě spokojenosti odměnit svou práci finančním darem.

TraderStar

Verze programu zdarma umožňuje základní analýzu dat, práci s indikátory, zobrazení novinek na burzovním trhu, ekonomický kalendář atd. Uživatelé mají možnost si napsat vlastní indikátory přímo v prostředí programu pomocí Java scriptu. Několik indikátorů je dokonce ke stažení na oficiálních stránkách prezentujících tento program. Aplikace běží na systémech MS Windows, Linux a Mac OS X.

ProRealTime

Bezplatná verze je omezená na použití pouze s EOD daty. Umožňuje technickou analýzu a práci s grafy. Výhodou je podpora mnoha operačních systémů (MS Windows, Linux, Mac OS X) včetně mobilních (Android, Windows mobile a Windows phone). V placené verzi programu jsou k dispozici realtime data a možnost obchodování s cennými papíry.

2.3 Komerční software

Jedná se o placený software s různou licenční politikou. Tato licence může omezovat počet současných instalací, přenositelnost či možnost modifikovat zdrojové kódy. Není to sice pravidlem, ale většinou tento druh softwaru nabízí oproti open source a freewaru větší a kvalitnější funkčnost.

AmiBroker

Z běžných funkcí nabízí práci s grafy, indikátory, upozornění apod. Oproti ostatním podobným aplikacím nabízí navíc 3D grafy, použití vrstev a vestavěný prohlížeč průzkumů. Pro tvorbu vlastních indikátorů a jiných rozšíření je k dispozici development kit. Mnoho indikátorů a rozšíření je k dispozici ke stažení na oficiálních stránkách. Indikátory a rozšíření se píšou v jazyce C/C++. Aplikace je dostupná pouze na OS MS Windows.

	Qtstalker	JStock	Grism	TraderStar (free)	ProRealTime (free)	AmiBroker
cena	zdarma	zdarma	zdarma	zdarma	zdarma	od 199\$
licence	GNU GPL	GNU GPLv2	GNU GPL	freeware	freeware	komerční
práce s grafy	•	•	•	•	•	•
vlastní indikátory	•	•	○	•	•	•
možnost upozornění	○	•	○	○	○	•
vlastní rozšíření/jazyk	•/C++	•/Java	•/Ruby	○/○	•/ProBuilder (podobný Basicu)	•/C/C++
real-time	•(plugin v rámci BP)	•	•	•	○	•
možnost obchodování	○	○	○	○	○	•
operační systém	GNU/Linux	MS Windows, Linux, Mac OS X	Linux/Unix, MS Windows	MS Windows, Mac OS X, Linux	MS Windows, Mac OS X, Android, Win Phone, Win Mobile	MS Windows

Tabulka 1: Srovnání aplikací pro práci s investičními nástroji

3. Zdroje dat a datová struktura

Pro svou funkčnost potřebují aplikace z oboru burzovního softwaru specifická data, které získávají pomocí webových služeb, importem ze souborů, popřípadě jinak. Data většinou obsahují pro specifický symbol informace, které lze vidět v tabulce 2 (uváděny jsou i anglické názvy, jelikož jde o zažité pojmy v burzovním oboru), popřípadě ještě jiné informace navíc v závislosti na dané službě. K dispozici je obecně několik typů dat.

První z nich jsou EOD data. Jedná se o souhrnné informace vázané k daným cenným papírům na konci burzovního dne. Druhým typem dat jsou realtime data. Tato data lze získat bezplatně, ovšem s určitým, byť krátkým zpožděním, například ze služeb Yahoo! Finance nebo Google Finance a taktéž se vztahují k jednotlivým cenným papírům.

Existuje také mnoho placených služeb, které poskytují úplné výsledky obchodování bez zpoždění. Těmto výsledkům se říká L1 (level 1) data. Navíc placené služby poskytují i tzv. L2 (level 2) data, které obsahují oproti L1 datům i informace o hloubce trhu, neboli objem nabídky a poptávky, díky nimž lze lépe určit budoucí vývoj daného investičního nástroje [3]. Dalším typem dat jsou různé statistiky, které se už však netýkají přímo konkrétních symbolů, ale spíše shrnují obchodování za celý burzovní den, popřípadě jiné časové období. Všechny tyto služby jsou poskytovány buď samotnou burzou, nebo prostřednictvím třetích stran.

Český název	Anglický název	Poznámka
ISIN	ISIN	Jednoznačný mezinárodní identifikátor cenných papírů
Název symbolu	Ticker	
Datum obchodování	Date of Exchange day též Trading day	
Čas obchodování	Trading time	
Minimální hodnota	Low	Minimální hodnota za daný časový interval
Maximální hodnota	High	Maximální hodnota za daný časový interval
Objem obchodů	Volume	Počet uzavřených obchodů za daný časový interval
Aktuální kurz	Trade value	
Změna	Change	Změna oproti předchozímu kurzu
Otevírající kurz	Open	Kurz, s nímž daná položka začíná v daném časovém intervalu
Uzavírající kurz	Close	Kurz, s nímž daná položka končí v daném časovém intervalu

Tabulka 2: Obecné informace získané pro jeden ISIN [3][4]

3.1 Burza cenných papírů Praha

Pražská burza poskytuje výsledky obchodování ve formě CSV souborů, které jsou k dispozici prostřednictvím FTP serveru. Tento server dovoluje paralelní stahování dat. Základní struktura složek na serveru je patrná z tabulky 3.

Název složky	Popis obsahu
Annual.rep	Výroční zprávy
aspnet_client	
cdip.dta	
Info.bas	Informace a články
Issuers.dta	Informace o emisích, denní soubory emitentů, popisy souborů
Press.rel	Informace pro tisk
Private.dta	
Results.act	Úplné výsledky obchodování
Results.ak	Kurzovní lístek se zpožděním
Results.old	Starší úplné výsledky obchodování
Semiannual.rep	Pololetní zprávy
Service.dta	
statist.dta	Denní, týdenní, měsíční, čtvrtletní a roční statistiky obchodování
Tom.dta	
Vestnik.doc	
VestnikCDCP.doc	

Tabulka 3: Struktura složek na FTP serveru pražské burzy v kořenovém adresáři (červeně jsou vyznačeny složky, které nejsou volně přístupné)

Na svých webových stránkách uveřejňuje pražská burza průběžné výsledky obchodování, ovšem se zpožděním 15 minut. Zdarma pak pomocí FTP poskytuje pouze výsledky z předchozího dne, jednoroční historii a statistiku. Úplné výsledky obchodování bez zpoždění jsou zpoplatněny a jsou poskytovány přes holdingovou skupinu CEE Stock Exchange Group. Tato skupina poskytuje data od devíti trhů. Pro informaci: roční poplatek za rok 2010 pro jedno připojení byl 1500 €, za L1 data bez zpoždění 3900 € a za L2 data 6300 € [6]. Výsledky obchodování za jeden den pro jeden ISIN obsahují informace, které lze vidět v tabulce 4 (uváděny jsou opět i anglické názvy, tentokrát tak, jak jsou přeložené pražskou burzou).

Název položky v češtině	Název položky v angličtině	Příklad možné hodnoty
ISIN	ISIN	CZ0005084459
Název CP	Name of security	"SM PLYNÁRENSKÁ"
BIC	BIC	"BAASMPY"
Datum burzovního dne	Date of Exchange day	"2012/04/17"
Závěrečný kurz	Closing official price	8874.00
Změna dnešní/předchozí	Change current/previous	0.00
Předchozí kurz	Previous official price	8874.00
Minimum za poslední rok	Minimum in the last year	7002.00
Maximum za poslední rok	Maximum in the last year	8874.00
Počet zobchodovaných CP	Trade volume (pcs.)	0
Objem obchodů	Trade value	0.00
Datum posledního obchodu	Date of the last trade	"2011/08/29"
Obchodní skupina	Trading group	"1"
Mód kotace	Type of listing	"2"
Trh	Market	"C"

Tabulka 4: EOD data získaná z pražské burzy pro jeden ISIN

3.2 Yahoo! Finance

Služby Yahoo! Finance poskytují informace z finanční oblasti. Tyto služby patří ke špičce ve finančním zpravodajství ve Spojených státech. Hlavním zdrojem je New York Stock Exchange, která je největší burzou akcií a derivátů na světě. Na této burze se obchoduje od 9:30 a.m. do 4:00 p.m. ET. Služeb Yahoo! Finance využívá většina bezplatných programů pro práci s investičními nástroji. Informace se získávají z CSV souborů, které služba vrací při speciálně formátovaném požadavku. Obecný požadavek na službu Yahoo! Finance:

<http://download.finance.yahoo.com/d/quotes.csv?s=NAZEVSYMBOLU&f=FORMATOVACIRETEZEC&e=.csv>

kde parametr za „s“ je požadovaný symbol a parametr za „f“ je řetězec složený z formátovacích symbolů. Význam jednotlivých symbolů je k dispozici v příloze A.

Příklad požadavku pro symbol aapl (Apple):

<http://download.finance.yahoo.com/d/quotes.csv?s=aapl&f=s1d1t1c1ohgv&e=.csv>

Vrácené informace spolu s vysvětlením použitého formátovacího řetězce jsou v tabulce 5.

Formátovací symbol	Význam for. symbolu	Příklad vrácené hodnoty
s	Symbol	AAPL
l1	Kurz posledního obchodování	607.26
d1	Datum obchodování	"4/18/2012"
t1	Čas obchodování	"11:53am"
c1	Změna kurzu oproti předchozímu	-2.44
o	Otevírající kurz	613.88
h	Denní maximum	620.25
g	Denní minimum	602.71
v	Objem obchodování	18835800

Tabulka 5: Význam použitých symbolů a návratové hodnoty pro požadavek v příkladu pro Yahoo! Finance [8]

3.3 Google Finance

Firma Google nabízí obdobné finanční služby jako Yahoo! Finance prostřednictvím své webové služby Google finance. Data poskytuje vývojářům pomocí Google Finance Portfolio Data API. Toto API je oficiálně zastaralé, ale podle vyjádření samotného Googlu není v plánu existující funkcionalitu rušit. Data jsou poskytována z mnoha burz, především ze spojených států. Bohužel data z českých burz nejsou v této službě k dispozici.

4. Burza cenných papírů Praha

Jedná se o nejvýznamnější burzu v České republice. Sídli v Praze a provozuje hlavní trh s akciemi v naší zemi. Veškerou její činnost kontroluje Národní banka. Ze zákona se jedná o akciovou společnost. Jelikož je burza založena na členském principu, mají přístup do burzovního systému pouze licencovaní obchodníci, kteří jsou pak zároveň členy burzy. Pražská burza je členem Federace evropských burz (FESE) a drží statut americké komise pro cenné papíry „Designated Offshore Market“, neboli bezpečný trh pro americké investory [7].

4.1 Burzovní den

Harmonogram burzovních dnů stanovuje burzovní komora vždy předem na jeden následující rok. Burzovní komora také stanovuje časový harmonogram burzovního dne pro dané období. Tento harmonogram lze vidět v tabulce 6. Burza zveřejňuje realtime data v kontinuálním režimu, který probíhá od 9:10 do 16:20 [7].

Čas	Obchodní režim
17:15 – 20:00	Aukce – sběr objednávek
17:15 – 20:00	Blokové obchody
17:15 – 20:00	SPAD – uzavřená fáze Obchodování se strukturovanými produkty – uzavřená fáze Obchodování s futures – uzavřená fáze
8:00 – 9:15	SPAD – uzavřená fáze Obchodování se strukturovanými produkty – uzavřená fáze Obchodování s futures – uzavřená fáze
8:00 – 16:20	Blokové obchody
8:00 – 8:45	Aukce – sběr objednávek
8:45 – 9:10	Aukce
9:10 – 16:20	Kontinuální režim
9:15 – 16:20	SPAD – otevřená fáze Obchodování se strukturovanými produkty – standardní otevřená fáze Obchodování s futures – otevřená fáze
16:20 - 16:27	Závěrečná aukce pro vybrané cenné papíry
9:11 - 16:27	Kontinuální výpočet indexu PX
17:15	Publikace kurzovního lístku
17:15 - 20:00	Download výsledků obchodování

Tabulka 6: Harmonogram burzovního dne [7]

4.2 Automatizovaný obchodní systém (AOS)

Jedná se o soubor zařízení, která umožňují uzavírání, registraci a vypořádání obchodů, elektronické zpracování dat a programové vybavení těchto zařízení. Součástí AOS je obchodní systém a systém pro vypořádání. Obchodní systém zajišťuje sběr objednávek nebo instrukcí, jejich následné párování a rozeslání informací členům burzy. Systém pro vypořádání slouží k zajištění vypořádání uzavřených obchodů nebo obchodů registrovaných v obchodním systému [7].

5. Seznámení s programem Qtstalker a ostatními prostředky

Pro instalaci a správný běh aplikace Qtstalker je nutné využít i několik dalších prostředků, bez kterých se aplikace neobejde. Jedná se o knihovnu pro technickou analýzu TA-Lib, databázi Berkeley a vývojářskou knihovnu QT. Pro pohodlnou úpravu zdrojových kódů, popřípadě vytvoření nových, je vhodné použít i vývojové prostředí QT Creator.

5.1 Qtstalker

Aplikace Qtstalker je vyvíjena komunitou uživatelů pod GNU GPL licenci. Projekt vede Steve Stratos. Qtstalker je primárně vyvíjen v prostředí QT Creator od Nokie. Z tohoto důvodu, jak je zvykem u mnoha linuxových aplikací, jsou první znaky názvu shodné s několika znaky v názvu vývojového prostředí, grafické knihovny nebo použitého programovacího jazyka. Poslední stabilní verze Qtstalkeru je 0.36, která využívá QT ve verzi 3.3. Tato verze Qtstalkeru je již zastaralá a z tohoto důvodu již ne-všechny pluginy fungují. Aktuálně se pracuje na verzi 0.37, která je dostupná pro vývojáře pomocí repositářů SVN, CVS a GIT. Qtstalker v čisté verzi obsahuje čtyři pluginy.

Yahoo plugin slouží pro stahování historických a zpožděných dat pomocí služby Yahoo! Finance. CSV plugin poskytuje uživateli možnost importu dat z CSV souborů. V nastavení před samotným importem musí uživatel vytvořit takzvané pravidlo, pomocí něž se data rozdělují na jednotlivé informace. Dále je umožněno nastavit automatický import z daného souboru v určitém časovém intervalu. To se hodí k suplování realtime funkčnosti, pokud má uživatel k dispozici jiný program, který provádí automaticky export dat do CSV souboru. NYBOT plugin slouží ke stažení EOD dat z NYSE. CME plugin poskytuje funkčnost pro stažení historických a EOD dat ze stejnojmenné burzy. Předchozí verze obsahovala i MySQL plugin, který byl ale odstraněn. Tento plugin je však stále k dispozici jako instalační balíček díky Marco van Zwetselaarovi.

Základní práce s aplikací je poměrně jednoduchá. Uživatel vybere požadovaný plugin z roletkového menu. Vyplní údaje, jako je časové rozmezí, hledané symboly, a stiskne tlačítko update. Následně může výsledné grafy zobrazovat pro různé časové období, nechat si zobrazit různé indikátory, přidávat vlastní značky do grafu, provádět exporty apod. Ukázka grafického uživatelského rozhraní je na obrázku 1.



Obrázek 1: Prostředí aplikace Qtstalker

5.2 Ta-Lib

Jedná se o knihovnu vyvíjenou od roku 1999 pro technickou analýzu finančních dat šířenou pod licencí BSD. Tato knihovna je dostupná pro programovací jazyky C/C++, Java, Perl, Python a jazyky .Net. Navíc je dostupná i ve formě doplňku pro MS Excel. Obsahuje přibližně 200 indikátorů. Aktuální verze je 0.4.0, přičemž Qtstalker vyžaduje minimální verzi 0.3.0. Mezi další programy, které využívají tuto knihovnu, patří například Advanced Stock Tracker, NextTick nebo EclipseTrader [9].

5.3 Berkeley-DB

Jedná se o databázi společnosti Oracle, která ji vyvíjí po získání společnosti Sleepycat. Aktuální verze je 11g. Dokumentace Qtstalkeru doporučuje nejnižší verzi této databáze 4.1 (4.3 pro uživatele FreeBSD), ale s novějšími verzemi od Oraculu již pravděpodobně Qtstalker nebude fungční. Původní databáze je psána v jazyce C a poskytuje API pro většinu programovacích jazyků. Existují ještě další dvě verze této databáze. Jedna je napsaná v jazyce Java a nabízena pod názvem Berkeley DB Java Edition. Druhá je napsána v jazyce C++ a nabízena jako Berkeley DB XML. Z aplikací které využívají tuto databázi je známý kupříkladu Evolution (funkcemi podobný aplikaci MS Outlook) obsažený v Gnome [10].

5.4 QT (knihovna)

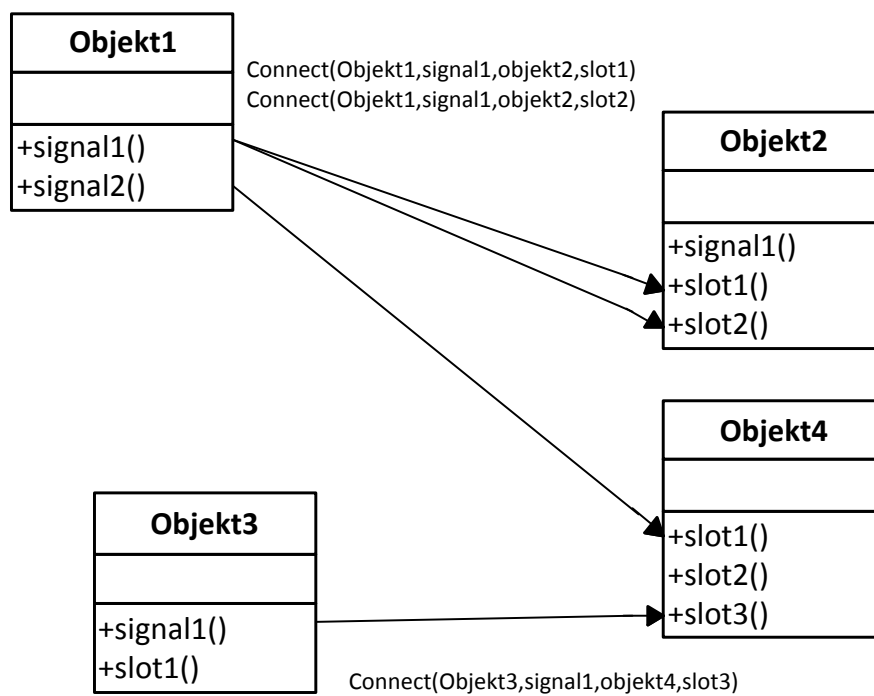
QT je vývojářská knihovna pro prostředí QT Creator. Též je označována jako QT development library nebo přímo společností Nokia nabízena jako Qt libraries. Knihovna usnadňuje práci s vlákny, databázemi, grafikou, síťovou komunikací apod [11]. Aktuální verze je 4.8.1 a je k dispozici pod licencí GNU GPL pro operační systémy MS Windows, Windows CE, Linux a Mac OS X. Qtstalker ve verzi 0.36 vyžaduje verzi QT 3.3. Mezi známé aplikace využívající této knihovny patří například VLC media player nebo VirtualBox.

5.5 QT Creator

QT Creator je vývojové prostředí od společnosti Nokia, které lze provozovat na operačních systémech MS Windows, Linux a Mac OS X. Toto prostředí slouží primárně k vývoji aplikací v jazyce C++. Integruje do sebe QT Designer pro tvorbu grafického rozhraní a je propojen s QT Assistant, díky němuž je přístupná kompletní dokumentace. Obsahuje simulátor pro mobilní uživatelské rozhraní. K debugování využívá debugger GDB a CDB. Pro přehled změn v kódu podporuje několik verzovacích systémů, mezi něž patří například CVS, Git nebo Subversion [11].

5.5.1 Signály a sloty

Signály a sloty jsou hlavním rysem QT a zajišťují komunikaci mezi objekty [2][11]. Všechny třídy, které dědí ze třídy *QObject* mohou využívat signály a sloty. Signál definovaný v jedné třídě emitujeme pomocí klíčového slova *emit* spolu se jménem signálu a předanými parametry. Tento signál můžeme v další třídě zachytit pomocí slotu, který připojíme k signálu pomocí statické metody *connect* ze třídy *QObject* [2]. Parametry signálu a slotu si musí navzájem odpovídat, to znamená, že pokud signál předává jako parametr hodnotu typu *int*, pak i slot musí přebírat jako parametr hodnotu typu *int*. Sloty se navíc chovají jako běžné metody a tudíž je možné je vyvolat i bez signálu. Způsob propojení několika objektů pomocí systému signálů a slotů je patrný na obrázku 2.



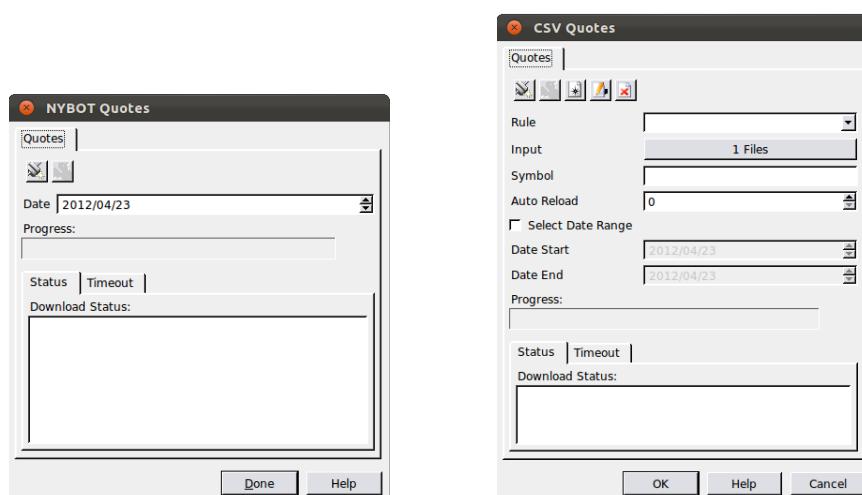
Obrázek 2: Způsob propojení několika objektů pomocí systému signálů a slotů

6. Obecná tvorba pluginu pro aplikaci Qtstalker

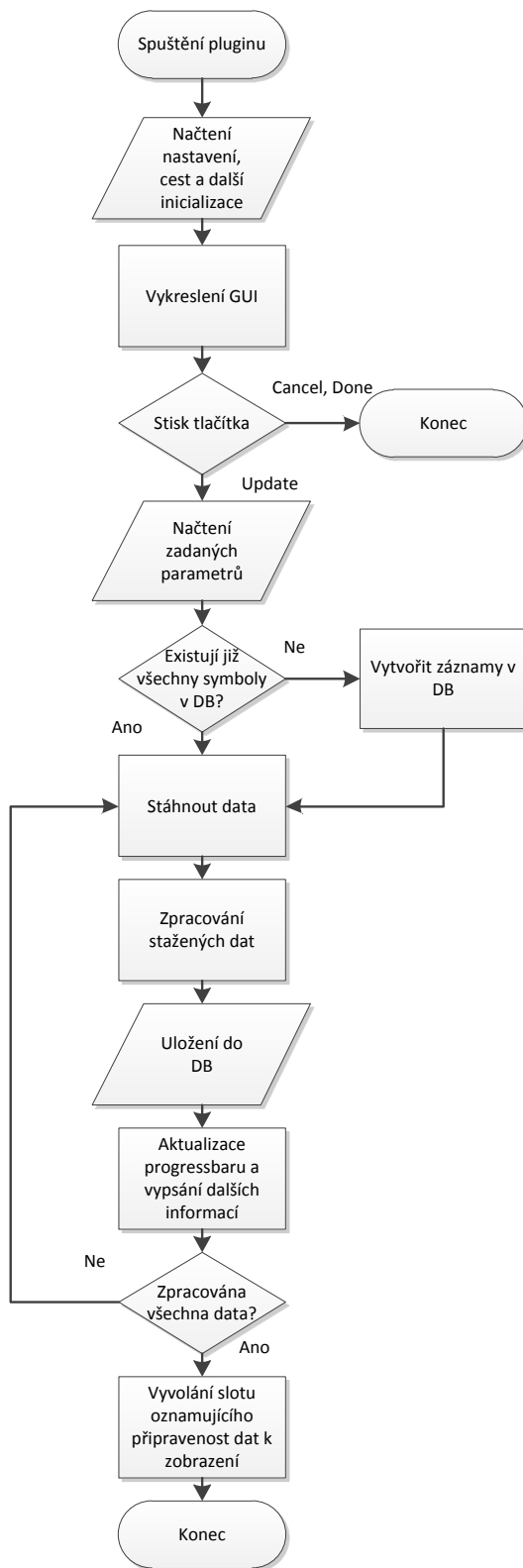
Pluginy pro Qtstalker lze vyvíjet v různých editorech, nicméně se doporučuje využít prostředí Qt Creator, které nabízí kompilaci jediným stiskem tlačítka, možnost debugování, našeptávač a mnoho dalších užitečných funkcí usnadňujících rychlý vývoj. Název pluginu by měl odpovídat způsobu získávání dat, nebo jejich zdroji, aby se uživatel mohl snáze zorientovat. Stěžejní pro tvorbu nového pluginu je rozhodnout se, jakým způsobem se budou data získávat a zpracovávat. To znamená, jestli budeme data získávat z webové služby, lokálně uloženého souboru apod. Data pak mohou být reprezentována různě. Například jako CSV, XML nebo obyčejný textový soubor. Ostatní funkční celky, jakými jsou ukládání dat do databáze, tvorba GUI apod., jsou u jednotlivých pluginů de facto stejné.

Přestože je tvorba GUI pro každý plugin téměř stejná, je důležité zjistit, jaké informace budeme pro získání dat potřebovat a tomu také přizpůsobit GUI. Některým pluginům například stačí pouze název symbolu, jiné zase vyžadují navíc přesné časové rozmezí, typ dat apod. Rozdíl je patrný z obrázku 3.

Pro všechny pluginy programu Qtstalker se dá popsat obecná funkčnost. Tato funkčnost je zobrazena pomocí zjednodušeného vývojového diagramu na obrázku 4. Po výběru pluginu v hlavním okně programu se vykreslí GUI daného pluginu. Po vyplnění požadovaných položek ve formuláři a stisku tlačítka update si plugin položky formuláře načte a popřípadě ještě zpracuje. Po té zkontroluje, zdali požadované symboly existují v databázi, aby k nim později mohl přiřadit data. Pokud symbol v databázi ještě neexistuje, pak se vytvoří. Následuje získání dat. Získaná data je nutné zpracovat do požadovaného formátu pro uložení do databáze. Po uložení se provede aktualizace progressbaru pro informování uživatele o vývoji stahování a zpracování dat. Pokud jsou všechna data zpracována, pak se z nich v hlavním okně programu vykreslí grafy. Během celého běhu pluginu se navíc zobrazují informace v textovém poli uživatelského rozhraní o důležitých událostech.



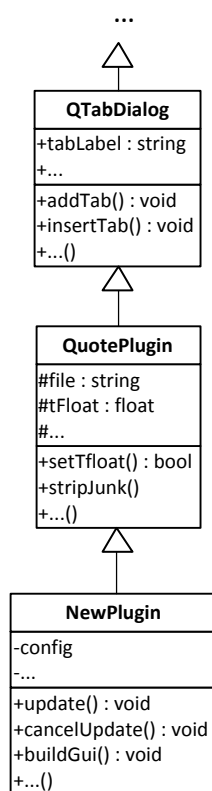
Obrázek 3: Rozdíl GUI dvou různých pluginů programu Qtstalker



Obrázek 4: Zjednodušený vývojový diagram funkčnosti pluginů programu Qtstalker

6.1 Seznámení s třídou QuotePlugin

Každý plugin Qtstalkeru dědí ze třídy *QuotePlugin*. Vztahy dědičností okolo této třídy lze vidět na obrázku 5. Třída *QuotePlugin* obsahuje veřejné metody a atributy, které lze vidět ve výpise 1. Z těchto metod je pro tvorbu funkčního pluginu nutné přepsat metody *Update* a *buildGUI*, pokud chceme přidat vlastní prvky do uživatelského rozhraní. Atribut *progressBar* slouží k zobrazení průběhu stahování dat nebo průběhu jejich zpracování. Metoda *setTFloat* slouží k převodu řetězce na datový typ float. Vrací hodnotu true v případě, že se převod zdařil a hodnota je pak uložena v atributu *tfloat*. Přebírá jako parametr řetězec určený k převodu a true hodnotu v případě, že chceme desetinnou čárku nahradit tečkou. V opačném případě bude desetinná čárka odstraněna.



Obrázek 5: Vztahy dědičností v okolí třídy *QuotePlugin*

```

1. QuotePlugin ();
2.     virtual ~QuotePlugin ();
3.     bool setTFloat (QString &, bool);
4.     void stripJunk (QString &, QString &);
5.     void createDirectory (QString &, QString &);
6.     void getPluginName (QString &);
7.     void getHelpFile (QString &);
8.     void buildGui ();
9.     void enableGUI ();
10.    void disableGUI ();
11.    virtual void update ();
12.    void setChartIndex (DBIndex *);
13.    QProgressBar *progressBar;

```

Výpis 1: veřejné metody a atributy třídy QuotePlugin

Dále třída obsahuje několik signálů (viz výpis 2). Ze signálů je nutné využít signál *signalWakeup*. Pokud využíváme prostředků pro stahování dat z této třídy, pak je nutné použít minimálně ještě signál *signalGetFileDone*.

```

1. void signalGetFileDone (bool);
2. void signalCopyFileDone (QString);
3. void signalTimeout ();
4. void signalProgMessage (int, int);
5. void chartUpdated ();
6. void signalWakeup ();

```

Výpis 2: signály třídy QuotePlugin

Sloty třídy QuotePlugin (viz výpis 3), tak jako jiné sloty programů vytvořených pomocí Qt Creatoru, se nemusí vázat pouze k signálům, ale lze je použít i jako obvyčejné metody. Pomocí *downloadComplete* oznamujeme, že plugin ukončil svou činnost a data jsou připravena k zobrazení. Vhodné je použít *printStatusLogMessage* pro výpis informací do okna pluginu. Ze slotů lze ještě využít *getFile* pro získání souboru ze sítě, pokud nepoužijeme vlastní způsob. Tento slot pak po dokončení stahování emituje signál *signalGetFileDone*, který připojíme k vlastní metodě, v níž stažená data zpracujeme.

```

1. void getFile (QString &);
2. void copyFile (QString &, QString &);
3. void getFileDone (QNetworkOperation *);
4. void copyFileDone (QNetworkOperation *);
5. void dataReady (const QByteArray &, QNetworkOperation *);
6. void slotTimeout ();
7. void getQuotes ();
8. void downloadComplete ();
9. void cancelDownload ();
10. void printStatusLogMessage (QString &);
11. void help ();
12. void slotWakeup ();

```

Výpis 3: sloty třídy QuotePlugin

6.2 Postup při vytváření pluginu

V prostředí QT Creatoru po otevření projektu Qtstalkeru přidáme nový podprojekt, popřípadě zkopírujeme jeden z již hotových pluginů a ten přejmenujeme a vyčistíme. Plugin je tvořen minimálně jednou třídou, která dědí ze třídy *QuotePlugin*. V naší zděděné třídě v konstruktoru si nastavíme atributy, provedeme další inicializace, nastavíme atributu *file* cestu k souboru s daty daného pluginu a atributu *dataPath* cestu k datům. Tyto cesty získáme ze třídy *Config* pomocí metody *getData(Config::identifikátor, QString data)*, kde *identifikátor* je označení pro data, která chceme získat, a *data* je proměnná, kde se následně data uloží. Pro *file* slouží identifikátor *Home* a pro *dataPath* identifikátor *DataPath*. Poté vyvoláme GUI pomocí metody *buildGUI*, kterou přepíšeme, pokud chceme přidat do uživatelského rozhraní vlastní prvky. Tělo konstruktoru lze vidět ve výpise 4.

```

1. NewPlugin::NewPlugin()
2. {
3.     qInitNetworkProtocols();//pokud využijeme pro stahování nadřazenou třídu
4.     //je nutná tato inicializace a následné připojení signálu do slotu
5.     connect(this, SIGNAL(signalGetFileDone(bool)), this, SLOT(fileDone(bool)));
6.     config.getData(Config::DataPath, dataPath);
7.     dataPath.append("/Stocks/NewPlugin");
8.     config.getData(Config::Home, file);
9.     file.append("/NewPluginDownload");
10.    //další inicializace
11.    buildGui();

```

Výpis 4: Konstruktor nového pluginu

V metodě *buildGUI* je nám umožněno přidávat prvky do GUI pluginu. Základní GUI pluginu již obsahuje tlačítka pro aktualizaci, storno aktualizace, tlačítko potvrzení a stornování pro celé okno, progress bar a textové pole pro výpis informací. Prvky můžeme přidávat libovolně, a to do mřížky na určitou pozici pomocí metody *addWidget*, kterou voláme na objekt *grid*. Po přidání všech požadovaných prvků je také vhodné použít metodu *resize* pro změnu velikosti okna do vhodných rozměrů viz výpis 5.

```

1. void NewPlugin::buildGui()
2. {
3.     setCaption(tr("NewPlugin Quotes")); //Popisek okna
4.     label = new QLabel(tr("Symbols"), baseWidget);
5.     grid->addWidget(label, 0, 0);
6.     QLineEdit *symbol = new QLineEdit(baseWidget);
7.     grid->addWidget(symbol, 0, 1);
8.     resize(400, 450);
9. }

```

Výpis 5: Tělo metody pro upravení GUI

Dále přepíšeme metodu *update*. Tato metoda je vyvolána po stisknutí tlačítka pro aktualizaci. V této metodě stáhneme data, zpracujeme, uložíme do databáze a vyvoláme signál *downloadComplete* oznamující, že data jsou připravena k zobrazení. Obecná těla metod *update* a *cancel* lze vidět ve výpise 6.

```

1. void NewPlugin::cancelUpdate()
2. {
3.     //zrušení stahování
4.     //nová inicializace atributů a proměnných
5. }
6.
7. void NewPlugin::update()
8. {
9.     //zjištění požadovaných symbolů
10.    //kontrola symbolů v databázi, jejich případné vytvoření
11.    getFile("nejaka url")
12.    emit signalWakeup();
13. }

```

Výpis 6: Těla metod *cancel* a *update*

Pro stažení dat máme dvě možnosti. Buď využít stávajících možností rodičovské třídy, nebo si napsat vlastní funkčnost. První možnost se hodí, pokud nám postačuje sériové stahování a zpracování dat. V případě, že bychom chtěli využít vícevláknové stahování a zpracování, musíme si tuto funkčnost napsat sami. Využijeme-li tedy stávajících prostředků rodičovské třídy, tak si vystačíme s metodou *getFile* přebírající jako parametr URL typu *QString* a signálem *signalGetFileDone*, který nám oznamuje, že data jsou stažena a jsou připravena v atributu *data* typu *QString*. Tento signál si připojíme na vlastní slot (viz výpis 7) pomocí metody *connect* [2].

```

14. void NewPlugin::fileDone(bool d)
15. {
16.     //parsování a další zpracování dat
17.     //uložení zpracovaných dat do databáze
18.     if(zpracována všechna data)
19.     {
20.         printStatusLogMessage(stringDone);
21.         downloadComplete();
22.         progressBar->reset();
23.     }
24.
25. }

```

Výpis 7: Tělo slotu pro odchytení signálu oznamujícího dokončení stahování

Po získání dat musíme tato data zpracovat, tzn. získat dílčí informace. Jako první nás bude zajímat datum a čas. Ten se do databáze ukládá ve formátu RRRRDDMMhmmss jako datový typ *QString*. Pokud konkrétní čas nelze získat, pak můžeme dané pozice nahradit nulami. Další informace se ukládají v datovém formátu typu *float* a jsou to následující položky: Open, Close, Volume, High a Low. Samotné ukládání se provádí tak, že otevřeme *DbPlugin plug* pomocí metody *open(path, chartIndex)*, kde *path* je cesta k aktuálně editovanému symbolu. Atributy *plug* a *chartIndex* jsou součástí rodičovské třídy a není tudíž nutné je vytvářet. Metoda pro otevření pak vrátí hodnotu *true* v případě, že se vyskytla nějaká chyba. V tomto případě zavřeme databázi zavoláním metody *close()* na atribut *plug* a ukončíme proces vkládání do databáze. V opačném případě vytvoříme proměnnou typu *Bar*, do níž uložíme data pomocí metod *setDate(QDate date)*, *setOpen(float)*, *setClose(float)*, *setVolume(float)*, *setHigh(float)* a *setLow(float)*. Nyní můžeme takto naplněnou proměnnou předat jako parametr pomocí metody *setBar(Bar bar)* databázovému pluginu. Databázový plugin poté zavřeme zavoláním metody *close()*. Příklad naplnění dat do databáze je vidět ve výpise 8.

```

1. plug.open("cesta k symbolu", chartIndex)
2. Bar bar;
3. bar.setDate("20120101123005");
4. bar.setOpen(203);
5. bar.setClose(203.1);
6. bar.setVolume(14555);
7. bar.setHigh(203.1);
8. bar.setLow(203);
9. plug.setBar(bar);
10. plug.close();

```

Výpis 8: uložení dat do databáze

Po zpracování všech požadovaných dat a jejich uložení do databáze zavoláme metody *printStatusLogMessage(stringDone)*, *downloadComplete()*, popřípadě ještě vyresetujeme *progressBar* pomocí metody *reset()*. Strukturu hlavičkového zdrojového kódu obecného pluginu, který využívá k získání dat prostředků nadřazené třídy, lze vidět ve výpise 9.

```

1. class NewPlugin: public QuotePlugin
2. {
3.     Q_OBJECT
4. public:
5.
6.     NewPlugin();
7.     void buildGui ();
8.     void update ();
9.     void cancelUpdate();
10.
11. public slots:
12.     void fileDone (bool);
13. private:
14.     //privátní atributy
15. };
16.
17. extern "C"
18. {
19.     QuotePlugin * createQuotePlugin ();
20. }

```

Výpis 9: zjednodušený hlavičkový zdrojový kód obecného pluginu

7. PSE plugin

Cílem tohoto pluginu je umožnit uživateli získat EOD data z pražské burzy a následně je zpřístupnit k dalšímu použití v programu Qtstalker. Jak již bylo zmíněno v kapitole 3.1, pražská burza cenných papírů poskytuje výsledky obchodování zdarma pouze ve formě EOD a uchovává podle vlastních informací jednoroční historii. V době tvorby této práce je však rozsah historických dat na FTP serverech pražské burzy více než dvouletý.

7.1 Funkčnost

Po stisknutí tlačítka update si plugin z dialogového okna vytáhne názvy požadovaných symbolů a časové rozmezí. Zkontroluje se, zdali jsou dané symboly již zavedeny v databázi. Pokud nejsou, pak se vytvoří. CVS soubory jsou na FTP serveru pojmenované podle data a pokud nejsou z aktuálního roku, pak jsou navíc v adresářích pojmenovaných podle ročníků. Proto bylo nutné si vytvořit seznam požadovaných souborů. Tento seznam obsahuje i soubory, které na serveru neexistují (díky víkendům, svátkům apod.). Poté se vytvoří několik objektů ze třídy *Downloader*. Těmto objektům se předává reference na vlastní plugin a název souboru, který mají stáhnout. Downloadery jsou na sobě nezávislé a jejich činnost probíhá paralelně. Každý downloader se připojí na FTP server pražské burzy, podle názvu souboru vyberou správnou složku a pokusí se soubor stáhnout. Poté emitují signál *signalGetFileDone*, kterým předávají informace o úspěchu či neúspěchu stažení dat a případná data ve formě textového řetězce. Tento signál je napojen na slot *FinishedDownload* ve třídě *PSE*, který data rozdělí na jednotlivé položky pro uložení do databáze. Po uložení do databáze se provede aktualizace progressbaru a zkontroluje se, zdali jsou všechna data již stažena. Pokud ne, pak se danému downloaderu předá další název souboru ke stažení. Pokud jsou všechna data stažena a zpracována, vypíše se informace o dokončení zpracování, vyresetuje se stav progressbaru a zavolá se slot *downloadComplete*. V tuto chvíli jsou data zobrazena pomocí grafu v hlavním okně Qtstalkeru a uživatel může okno pluginu zavřít, nebo plugin znovu použít.

7.2 Zpracování dat

Data jsou z FTP serveru pražské burzy získávána ve formě CSV souborů. Na jeden burzovní den připadá jeden CSV soubor historie. Pro každý ISIN jsou hodnoty na jednom řádku. Hodnoty jsou odděleny čárkami. Jelikož ale některé položky obsahují i čárky v názvech, nelze použít pro jednoduché rozdělení těchto hodnot pouze metodu *split* ze třídy *QString*, která vrací list rozdělených hodnot, ale je nutné nejprve nalézt čárky v řetězcích a ty nahradit jiným

symbolem. Naštěstí jsou řetězce označeny uvozovkami, takže je možné projít každý řádek a provést kontrolu vždy mezi dvěma uvozovkami. Před samotným rozdělením řetězce je ještě vhodné odstranit nadbytečné mezery a lomítka. Poté je možné provést rozdělení řetězce. Datum je nutné doplnit o řetězec šesti nul, aby splňovalo požadovaný formát pro uložení do databáze. Pražská burza navíc neposkytuje v EOD datech položky maximum a minimum pro daný burzovní den, ale hodnoty za celý rok. Z tohoto důvodu jsou do databáze uloženy na místo hodnot maximum a minimum hodnoty open a close podle toho, která z nich je větší.

7.3 Downloader

Pro zajištění vícevláknového stahování a zpracování dat bylo nutné vytvořit novou třídu, která by zajistila stažení požadovaného souboru a upozornila na dokončené stažení. Downloader využívá třídu *QFtp*, která je nadstavbou třídy *QNetworkProtocol* a usnadňuje práci s protokolem FTP. Po připojení na FTP server pražské burzy je možné procházet jednotlivé složky a stahovat data do objektů, které dědí ze třídy *QIODevice*. V tomto případě je využit objekt třídy *QBuffer*. Dokončení stažení dat pomocí *QFtp* lze zjistit díky signálu *commandFinished*. Zde je však nutné porovnat id příkazů, jelikož tento signál je emitován po každé dokončené operaci (např. připojení, přihlášení atd.). Na tento signál je navázán slot *FinishedDownload* (jiný než ve třídě *PSE*). V tomto slotu se odchytlí případné chyby, data z bufferu se převedou na řetězec a předají signálem *signalGetFileDone* pluginu. V downloaderu je dále použit časovač pro případ, že by FTP server příliš dlouho neodpovídal. Tento časovač je pak napojen na slot, který opět emituje signál *signalGetFileDone*, tentokrát ale s oznámením o chybě.

7.4 Problémy během vývoje

Při koexistenci verzí knihovny QT 3.3 a 4 nabízí našeptávač pouze možnosti z verze 4. Stejně tak integrovaný prohlížeč dokumentace nabízí dokumentaci pouze pro novější verzi. Je tedy nutné mít otevřenou dokumentaci pro verzi 3.3 ve webovém prohlížeči.

Třída *QThreadPool* pro recyklaci vláken je obsažena až od verze QT 4. Při vývoji byly vyzkoušeny různé pooly vláken z externích knihoven, ale běh Qtstalkeru vždy skončil chybou oznamující chybu v daném pluginu.

8. YahooRealtime plugin

Tento plugin slouží pro průběžné sledování hodnot cenných papírů pomocí služby Yahoo! Finance. Pro získání dat je použit stejný formátovací řetězec jako v příkladu v kapitole 3.2. Informace zobrazované díky tomuto pluginu ve skutečnosti nejsou reálné, ale jsou díky vlastnostem služby Yahoo! Finance zpožděny. Na rozdíl od PSE pluginu je využito stahování pomocí metod nadřazené třídy, jelikož data není potřeba stahovat paralelně.

8.1 Funkčnost

Jakmile uživatel stiskne tlačítko update, plugin si zjistí požadované symboly. Tyto symboly v databázi nejsou, jelikož se databáze pro tyto symboly používá jen po dobu spuštění Qtstalkeru a po jeho ukončení jsou z databáze odstraněny. Z tohoto důvodu se symboly v databázi rovnou vytvoří. Data jsou pak stahována v intervalech nastavených uživatelem. Tyto intervaly se řídí pomocí timeru. Pokaždé, když timer vyšle signál, se v cyklu stáhnou data pro všechny požadované symboly a zpracují se. Následně se timer vyresetuje a znovu spustí. Po každém úspěšném zpracování se volá slot *downloadComplete*, to znamená, že se graf průběžně aktualizuje. Uživatel může dialogové okno pluginu zavřít a stahování bude dále pokračovat.

8.2 Zpracování dat

Data lze jednoduše rozdělit pomocí čárky. Jediné položky, které vyžadují úpravu, jsou datum a čas. Datum opět rozdělíme pomocí lomítek a jednotlivé části upravíme. Úprava spočívá v přidání nuly na začátek, pokud se jedná o část tvořenou jediným symbolem (hodnota data nebo měsíce menší než 10). Čas je ve 12 hodinovém formátu s postfixem am nebo pm. Je nutné jej tedy převést na 24 hodinový. Následně se datum a čas složí do požadovaného formátu pro uložení do databáze.

9. Závěr

Díky této bakalářské práci jsem získal větší zkušenosti se systémy unixového typu a vývoji svobodného softwaru pro tuto platformu stejně tak, jako jsem nabyl lepších znalostí v burzovním oboru.

Bylo nutné překonat některá úskalí v podobě nekompatibilních knihoven, nutnosti pracovat podle dokumentace vystavené na webu a chybějícího našeptávače. Naštěstí i přes absenci komentářů v původních zdrojových kódech, je kód poměrně dobře čitelný a pochopitelný. Navíc mnoho důležitých informací ohledně chyb při sestavování projektu bylo možné nalézt na oficiálním fóru [5].

Oba pluginy vytvořené v této práci jsou funkční a reálně použitelné. Jediná vážnější chyba, na kterou jsem narazil při jejich tvorbě, bylo nefunkční tlačítko pro stornování probíhajícího updatu. Tato chyba se projevuje u všech pluginů a je způsobena nedostatkem v rodičovské třídě. Navíc některé pluginy již nefungují, jelikož verze Qtstalkeru 0.36 je již poměrně zastaralá.

V budoucnu by bylo vhodné odstranit nedostatek ohledně cancel tlačítka. Dále by se aplikace dala rozšířit o možnost skutečného realtime obchodování. Tzn. umožnit uživateli například přes nějakou placenou službu sledovat vývoj na burze a umožnit mu nákup či prodej cenných papírů. V době tvorby této bakalářské práce se pracuje na verzi Qtstalkeru 0.37, takže je možné, že program dozná takových změn, které se dotknou i způsobu tvorby nových rozšíření a pluginy v této práci bude nutné přepracovat pro vyšší verzi.

10. Reference

- [1] Prata Stephen, Mistrovství v C++, COMPUTER PRESS, 3. aktualizované vydání 2007, ISBN 978-80-251-1749-1
- [2] Blanchette Jasmin, Summerfield Mark, C++ GUI Programming with Qt 3, Prentice Hall, 2004, ISBN 978-0131240728
- [3] Anatoly B. Schmidt, Financial Markets and Trading: An Introduction to Market Microstructure and Trading Strategies, Wiley Finance, 1 edition, 2011, ISBN 978-0470924129
- [4] Štýbr David a kolektiv, Začínáme investovat a obchodovat na kapitálových trzích, GRADA, 20011, ISBN 978-80-247-3648-8
Dostupné z <http://books.google.cz>
- [5] sourceforge, Qtstalker help forum [online] 2012 [cit. 2012-04-12]
Dostupné z <http://sourceforge.net/projects/qtstalker/forums/forum/80701>
- [6] Ceník burzovních informací [online] 2010 [cit. 2012-04-16]
Dostupné z ftp://ftp.pse.cz/Info.bas/Cz/cenik_informaci.pdf
- [7] Prague Stock Exchange [online] 2012 [cit. 2012-04-16],
Dostupné z <http://www.pse.cz>
- [8] MDBitz Development Resources [online] 2009 [cit. 2012-04-14],
Dostupné z <http://resources.mdbitz.com/2010/02/understanding-yahoo-finance-stock-quotes-and-s11d1t1c1ohgv/>
- [9] Ta-Lib [online] [cit. 2012-04-16],
Dostupné z <http://ta-lib.org/>
- [10] Berkeley DB [online] 2012 [cit. 2012-04-16],
Dostupné z http://en.wikipedia.org/wiki/Berkeley_DB
- [11] QT [online] 2012 [cit. 2012-04-16],
Dostupné z <http://qt.nokia.com/>

A Tabulka parametrů pro požadavek na službu Yahoo! Finance

Variable	Description	Variable	Description	Variable	Description
a	ask	g4	holdings gain	n4	notes
a2	average daily volume	g5	holdings gain percent real time	o	open
a5	ask size	g6	holdings gain real time	p	previous close
b	bid	i	more info	p1	price paid
b2	ask real time	i5	order book real time	c2	change in percent
b3	bid real time	j1	market capitalization	p5	price per sales
b4	book value	j3	market capitalization real time	p6	price per book
b6	bid size	j4	EBITDA	q	ex-dividend date
c	change and percent change	j5	change from 52 week low	r	price earnings ratio
c1	change	j6	percent change from 52 week low	r1	dividend pay date
c3	commision	k1	last trade with time (real time)	r2	price earnings ratio real time
c6	change real time	k2	change percent real time	r5	PEG ratio
c8	after hours change real time	k3	last trade size	r6	price per EPS estimate current year
d	dividend per share	k4	change from 52 week high	r7	price per EPS estimate next year
d1	last trade date	k5	percent change from 52 week high	s	symbol
d2	trade date	l	last trade with time	s1	shares owned
e	earnings per share	l1	last trade	s7	short ratio
e1	error indication	l2	high limit	t1	last trade time
e7	EPS estimate current year	l3	low limit	t6	trade links
e8	EPS estimate next year	m	days range	t7	ticker trend
e9	EPS estimate next quarter	m2	days range real time	t8	1 year target price
f6	float shares	m3	50 day moving average	v	volume
g	days low	m4	200 day moving average	v1	holdings value

h	days high	m5	change from 200 day moving average	v7	holdings value real time
j	52 week low	m6	percent change from 200 day moving average	w	52 week range
k	52 week high	m7	change from 50 day moving average	w1	days value change
g1	holdings gain percent	m8	percent change from 50 day moving average	w4	days value change real time
g3	annualized gain	n	name	x	stock exchange
				y	dividend yield

B CD

CD obsahuje adresáře:

- | | |
|----------|--|
| Software | - Zdrojové kódy aplikace, potřebné knihovny, soubor README.txt s dalšími informacemi |
| Text | - Text bakalářské práce |